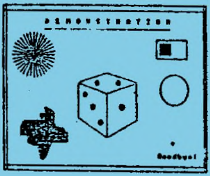
ISSN 0112-3351

BEEBLET

CONTENTS

Editorial Notes 2
Around the Regions 3
Of Modems,User Guides and Speech 6
HELP 8
Great News! 9
Electron Column 10
Classified Ads 11,36
Spy 2 - Review 12
Drawing Program (part 1) 14
Stacking Coins 22
Adventure Games 25
Adder (listing) 30
Elope! - with those envelopes 32
Look - No Files! 34
Saving Screens 37
Macsim (listing) 38
Hall of Fame 39
Local Groups 40



VOLUME 3 NUMBER 8 SEPTEMBER 1985

BBC / ACORN

COMPUTER USER GROUP NZ (inc)

↓↓

EDITORIAL NOTES

CORRECTION - The Acorn Saga

Mr Douglas Pauling of Barsons has kindly supplied the following update:
"Olivetti in fact have increased their stake in Acorn Computers to 70 percent and a new Managing Director has been appointed Mr. Brian Long ex Hawker Siddley Canada. Apart from one Olivetti man there is no major influx by Olivetti or AT&T executives.

Under the scheme of arrangements made with the creditors and the new funding made possible by Olivetti Acorn is posed to do great things in the future."

COMPETITION

Entries for the Compettion announced in the April Beeblet have now closed off. I will be gathering up a selection of programs submitted between April and now, published or not, and forward them to an umpire chosen by the National Committee.

The Competition seems to have stimulated quite a reasonable level of activity (together with the odd goad..) so think about some other interesting forms of promotion.... Any ideas (sponsors?).

COMING...

3D Text (and other Neville T. programs), Logo (article and examples), Submarine Hunt game, many reviews, listings, comments, President's report on future options, Help, tips, articles for beginners......

ADVERTISERS:

Pitman Publishing N.Z.	5
Modernage Instruments Ltd.	13

****

BEEBLET is edited by Bruce Wills who may sometimes be tracked down on 861-662 (h) or 729-929 x8896 (wk).

DISCLAIMER:

The views of the reviewers, the Editor and other contributors are not necessarily shared by the
BBC/ACORN COMPUTER USER GROUP NZ (INC)

↑↑

AROUND THE REGIONS

THE AUCKLAND REPORT

Steve Williams

Last meeting was well attended, with the topic being expansion options i.e. Solidisk vs Aries etc. Not as many questions as expected, maybe most people have already made their mind up... Next meeting is all about printers. A reminder to all Auckland members that there will shortly be several vacancies on the Auckland committee. AGM is planned for the meeting after next (i.e. October).

WELLINGTON GROUP

WORKSHOPS:
These are held 2nd Thursday of each month at Wellington College. There is a network of BBCs available. They were instituted to partly reach those members who cannot attend the Wednesday meetings, but few seem to be taking this opportunity. Because we have to pay a substantial amount for the machines, there is a $2.00 per adult door charge.

MEETING TOPICS:
Sept 12 Thurs dual programs ie for more than one 'player'
25 Weds Word processing, from beginning, to a brief expose on each of the ones currently available.

Oct 10 Thurs try using VIEW at this workshop - it's in every machine.
30 Weds using discs, and Disc filing including Random Access files

Nov 14 Thurs adventure systems
27 Weds Music, hopefully the Music 500 will be put through its paces.
A picnic is being planned for the January meeting.

HOUSE GROUPS
A sheet is on the notice board at Wednesday meetings, or contact one of the committee members if you are interested in these. The idea is that a few people can gather in a neighbour's home (or their own) to talk 'shop'. You may have noticed the odd article in Beeblet coming out of informal workshops held in Hataitai...

* * *

WAIKATO

The 2 September meeting featured a hardware session and discussions on BASIC, assembler and Pascal programming, together with a few games (school holidays).

'Waibug' notes that meetings are held in Room E112 of Block E of Waikato Tech - enter via Gate 7 (opposite Girls high in Tristram Street) and its the new four floored block on your right.

N.Z. FORCE S.E.A. BBC COMPUTER USER GROUP

Flt Lt Gary Bennison:
There is a thriving user group, mainly of NZ Forces personnel, but with 5 civilian members. The group sells discs, eproms, and has an extensive software library available for loan to members. We meet once a month informally and discuss a set topic. A lot of the members have very little experience of computing so we take a lot of time and explain the hardware and useful software packages. The sale of computers is very competitive here with Apple II and IBM-copies selling for less than the BBC (BBC B+ S$1700), so the User Group has to actively promote the "Beeb" to ensure that people are aware of its potential and are not swayed away from it by the cheaper prices of the other computers.

We have had a troubled time over the last 8 months as the BBC agency here collapsed and there was no hardware or software available for that period. However there is a new agency here who are making a real effort to push the machine. In fact they cannot keep up with the demand at the moment. There is no outlet in Singapore that sells BBC software at the moment. We have to import it all from the UK.

I found your publication BEEBLET very interesting with lots of good programs and tips.

If any of your members are passing through Singapore feel free to contact me.

ACCESS RADIO

Broadcasts at 9 am on a Saturday morning (2YB, 783 Khz). Once a month BBC/Electron programs are broadcast. Use your tape recorder plugged in to the radio/amplifier to record the programs, then plug it into the computer to load them.

BBC PUBLICATIONS

BBC 'MASTER GUIDES' SERIES

MASTERING PROGRAMMING

Aimed at the dedicated BBC Micro user, this books looks in detail at machine codes and assembly language, as well as giving an in-depth analysis of BBC Basic. For the serious (or would-be serious) programmer - nearly 50% of the book's pages show listings of the subjects described.
Approx $39.95 Due late 1985

MASTERING MUSIC

A full grounding in how to learn and play both simple music melody lines to highly sophisticated arrangements such as "The Ride of the Valkyries". A very comprehensive music tutorial for understanding and making music which gives examples both in computer listing form as well as in music notation.
Approx $39.95 Due late 1985

Also available shortly:

TOOLBOX II (BOOK)

A collection of 25 utility programs designed to enhance the capabilities of the BBC microcomputer and Acorn Electron. The utilities fall into five groups: Basic Programming, Assembly Language Programming, Graphics, Filing System Aids, Miscellaneous. There is accompanying software which will run from tape or disc.
Approx $39.95

PITMAN PUBLISHING NEW ZEALAND
PO BOX 38-688
PETONE

OF MODEMS, USER GUIDES, AND SPEECH

Steve Williams Auckland

COMPUTER CONCEPTS SPEECH ROM

To use this ROM you need to have at least the TMS 5220 speech chip fitted in IC99. It grabs two pages of memory (i.e. puts PAGE to 1B00 on DFS machine). Very simple to use, and the quality of speech is excellent, definitely the best way to go for speech synthesis.

ADVANCED DISK USER GUIDE

This excellent book, by Colin Pharo is available mail order from UK. (See Ad in Beebug or Acorn User). A bit pricey, but well worth it. More of a reference book than a read-cover-to-cover. Invaluable for both expert and novice.

BASIC ROM USER GUIDE

By Mark Plumbley
Applies to both BBC and Electron. Allows the novice machine-code programmer to learn a few "tricks" as well as explaining how BASIC works. Quite a few utility programs printed in the magazine (I also have the accompanying software). Another good buy.

MODEMS

By the time you read this, this information may be out of date, but modems currently available, and their status is as follows:

ACORN - Prestel adaptor, 1200/75 only. Price about $600 including ROM software. Has PO approval

ASMAIL - Attache systems. 1200/75 only (300/300 appears available but is disabled to get PO approval by bending one of the IC pins out.....) Price $400 is very reasonable. Has been PO approved.

NZ MICRO CLUB - 1200/75,75/1200,300/300 The "Ray James" modem. Price "under $200" as a KIT. More if assembled. I can vouch for this unit, having built one myself. It WORKS! Mains-powered. NOT PO approved, but will be submitted for approval shortly.

WELLINGTON BBC GROUP - 1200/75, 300/300. The "Neil Gordon" modem. Price "under $200". Prototype working, no more information available as yet. (Ed.'s note: I understand that P.C. boards have been manufactured and are currently being populated with a view to submitting a modem for P.O. type approval in the near future. The modem will come boxed with plugpack ie. effectively mains powered.)

WHAT YOU CAN DIAL

Locally, there is not much available at the moment. ADITEL (Prestel clone) is the pick of the commercial stuff, but at $180 per year it's obviously not interested in the private individual. NZ Micro Club in Auckland have their own BBS, running on a Stride (300/300 8-N-1, dial 397748, user COMMON, nil password). I understand there is one running in Wellington too, but no details known (Ed.'s note : currently down while Bryce Francis bring up their printing package - the reason they paid out some $1/4 million for their VAX... a second telephone line has however been ordered for a certain Evans Bay house ....) Auckland User group is setting up a BBS at Barson's, but currently waiting on the Post Office.

Through PACNET, the world is open to you, (albeit at approx NZ$25 per hour). I currently subscribe to the SOURCE (US) and Telecom Gold (UK). Also play MUD on Essex machine sometimes. Favourite pastime is CHATting on the SOURCE, quite a buzz! Their member directory also lets you meet other people with similar tests (located 78 people interested in Adventure games, 8 with BBC's etc....). Definitely worth a try.

GETTING STARTED WITH COMMS

Anyone who would like advice on modems, comms, how to subscribe to PACNET, databases etc please feel free to contact me at PO Box 6662, Wellesley Street, Auckland or ring AK 4446529.

QUESTIONS

- Does anyone have, or know of a terminal emulation program or package that will emulate a Data General D200 or D210 terminal???
- Why hasn't anyone written a ROM that will check BASIC lines for syntax AS YOU ENTER THEM. Surely it can't be that hard....

****

HELP

A forum conducted by Tim Ryan

Hi, not many people wanting to know anything this time. I guess you all know everything there is to know about your BBC by now. One or two questions, and one or two ideas.

1. J. Logie writes to point out that in the Beeb user guide there is yet another mistake to add to your collection of corrections:
On page 489 character 228 is shown wrongly. It should actually look like character 100 on page 488.

2. Beryl D of Wgtn asks: Does anyone know the constant needed to start the Beeb's random number generator off at the point it starts at switch on (using x=RND(-value)

Well I don't offhand, but I'm sure you could write a program to find out.
Here's a start - it may not work as is, but you should be able to adapt it.

```
  1 REM first get the numbers to check against
 10 DIM T(20)
 20 FOR X=0 TO 20
 30   T(X)=RND(20)
 40   NEXT
 45 REM now do the compare
 50 FOR X%=0 TO -(2^32-1) STEP -1
 60   dummy=RND(X%)
 70   REPEAT
 80     test=RND(20)
 90     UNTIL test<>T(count) OR count=20
100   IF count=20 THEN PRINT "The answer probably is ";X%:END
110   NEXT
```

By the way, the program as is, will probably take several months to run, if the number is a big one, so I suggest you start it off on a long weekend once you are sure the bugs are out.

↓↓↓

3 A standard 'mistake' I've noticed in some of the programs coming in for the magazine is to put the following:

```
10 MODE x
20 CLS
30 .....
```

The MODE command clears the screen itself so there is no need to do a CLS as well. I imagine what happened is that originally the programmer only had the CLS and added a MODE command later forgetting to remove the CLS.

↓↓↓

GREAT NEWS!!

A Half-year Membership

As of October, potential members can be offered two options:

1. Full year membership as previously

2. Half_year membership:

PAY $15.00^

(^ plus local levy if applicable)
A subscription will start with the October issue of Beeblet, and expire with the March 1986 issue.

If you join for the half year only you will not be able to get back issues for earlier this year (ie April - September '85) except at the cost of $3.00 each.

SPREAD

THE GOOD NEWS!!

↑↑

ELECTRON COLUMN

Steve Cosgrove

One of the more practical changes made in the Electron is the key marked FUNC. While the cynic could claim that it replaced 10 keys across the top of the keyboard, I perfer to claim that it added an extra 47 or so keys!

Those who have used VIEW on the BBC will be know the frustration apparent when some commands are accessed by <Function Key>, others by <CTRL + Function Key>, or <SHIFT + Function Key>. It was a pain, I was always getting confused. On the Electron, all commands are accessed by <FUNC + key>, which is very much easier to handle. (Or finger as the case may be.)

These are referred to as "firm" keys. This is in between "hard" keys, which are ABCDEFGHI...., and "soft" keys, which are the standard "programmable" keys - see chapter 24, EUG. Basically, they are not as flexible as "soft" keys, and are more flexible than "hard" keys.

To use <FUNC + key>, investigate page 283, EUG. Using an *FX call 225, 226, or 227, with a value larger than 1 disables the BASIC keywords printed on each key, and causes the key to output an ASCII code instead. The EUG seems to have a line missing. It refers to "the two (values) given above". These values are not there, as far as I can see.
This is how the *FX calls affect FUNC keys.
0 causes the keys to have no effect.
1 causes the keys to output its normal BASIC keyword, or definition.
2 - 255 causes the keys to output ASCII codes.
To use an example, define the User-defined keys with something, then try:-
*FX225 0 Disables the User-defined keys
*FX225 65 Causes User-defined keys to output "ABCDEFGHIJ". ASCII values 65 to 74 respectivly
*FX225 1 Causes the keys to output their definitions as normal
*FX226 affects A to P, and *FX227 affects Q to Z. Instead of user definitions, these keys output their BASIC keywords.

On the BBC, these commands control Function Key, SHIFT + Function Key, and CTRL + Function Key. *FX228 also controls SHIFT + CTRL + Function Key. Fortunately, this option does nothing in the Electron

↓↓

By using 224 as a base for these keys, the standard user-definable characters can be accessed directly from the keyboard. Page 93ff, EUG, explains how to redefine these characters. After typing in, running, getting sick of, and ESCAPEing, the program on page 95, type *FX226 224 <RETURN>. Then hit FUNC A, and FUNC B.

Have fun, play with these keys, find lots more questions, and send them to me, C/- the User Group.

In Reply. Steve S, Auck

Thank you for your letter. Note the page of "Jargon 2" in next issue, dealing with printer drivers. The Super 5 printer is Epson compatable. It is, in fact, a Logitec, with a different name. I have a VIEW printer driver, which I use all the time on my own Super 5 printer. (Great minds...) You can expect a copy in the mail, together with full details on how to use it. Your problems with extra space at the top and bottom of each page is not unique. My Super 5 gives up trying when it has 9 lines to go. Pushing the "On Line" button, however, will print another line, and you can do this about 5 times without the paper slipping. After much practice, I can print right down to the bottom of the page, hitting On Line for each line, and moving the paper up by hand if necessary. For typing single page letters in VIEW, I normally use the "PB 0" embedded command. It should suppress all margins, top and bottom. Otherwise set each of the following to 0: TM, HM, BM, & FM. Thank you again for your letter. Please let us know how you get on.

↓↓

CLASSIFIED ADS

****

DEALS AND DISCOUNTS

Send stamped addressed envelope for details of the deals negotiated by the UG for members.

FOR SALE:

****

BBC system (2 yrs old), 12" Zenith Green Screen, games, books and magazines (including 2 year's worth of Beeblets), blank tapes etc.

$1,700 ono. Ph. Andrew Christie, (Wgtn) 673-131.

↑↑

SPY 2 - 8K UTILITY ROM FROM SYSTEM SOFTWARE

Steve W. Auckland

This ROM comes from the firm which produced ADE, the definitive machine code programmers aid and in fact includes many features resident in ADE. However as well as the various debugging, and disassembling commands there are also a batch of disk utilities.
The machine - code monitor side of SPY enables you to do the following:

- Set Break points in machine-code (up to 8 simultaneously)
- Displays current status of PC, X and Y registers, Accumulator, Status flags and Stack pointer.
- Edit memory directly
- Examine memory in ASCII, disassembled or Hex format
- Single-step through code, displaying status with each step.
- Modify program counter dynamically.

All in all there isn't much you can't do!. There are a lot of different commands, so it takes a while to get the hang of using them but I think that both novice and expert m/c programmers will soon wonder how they managed without it.
The disassembler has two options available, the most useful of which copes with embedded data and displays it separately. Output can be to screen or printer and OS addresses such as OSBYTE,OSWORD etc are converted to enable easier reading i.e. JSR &FFF4 becomes JSR OSBYTE.

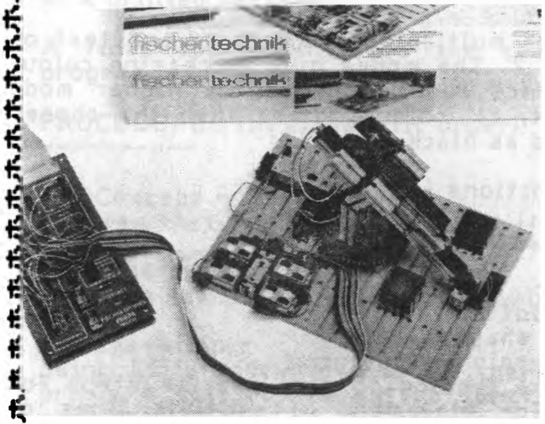
The disk utilities are very similar to Disk Doctor. There is a disk sector editor,unfortunately in 80-column mode only. A command to amend a files load or execution address, a formatting command, a command to make a directory entry for a file on disk (useful when recovering files deleted accidentally!), a load - then - move - down - in - memory command, a reformat command which reformats a particular track, then writes the data back (as well as it can), a status command which gives information about the disk such as number of tracks, and the location (if any) of any "gaps" (gaps will appear where a file is deleted and a *COMPACT is not done), a verify command which will check readability of each sector, leaving data intact, a command to search a disk for occurrences of a particular string.

Another useful command provided is one which prints all sideways ROM's installed and the sockets they reside in.

SUMMARY:At 25 pounds (approx $60), this is not cheap, and the absolute novice will get little use from it,however for anyone contemplating assembler programming, or needing the disc utilities it provides SPY2 is certainly a better bet than Disk Doctor and can be strongly recommended.

modernage instruments limited

ROBOTIC and COMPUTER INTERFACING SPECIALISTS



Fischertechnik robotics kit and UNILAB interface for BBC now available for $450

For your copy of our catalogue and price list write to PO Box 31-096 or phone 585-508 Christchurch

DRAWING PROGRAM

by Bill T.

OBJECTIVE:

I wanted to have a screen drawing program with the ability to print the drawing area onto an 80-column printer.

A couple of programs which were keyed in from magazines were both below my expectations. However, having learned from this experience, I decided to write my own. The result to date is presented here. I say "to date" because I find that new features are being added as I use the program for various different tasks.

PROGRAM SUMMARY

The system consists of 2 programs:-

T-DRAW which prints instructions and a keystrip for the function keys

T-DRAW2 which is the main drawing program and is chained from T-DRAW

The program T-DRAW2 allows multicoloured graphics and text on the screen in modes 0, 1 or 2. The modes give increasing colour choice but decreasing graphics resolution for the higher mode numbers. For printed output, all colours other than the chosen background colour are printed as black.

Features include built-in functions for:

- 3 kinds of circles (or partial circles)
- 2 kinds of rectangles
- solid squares of colour
- line drawing (solid or dotted)
- colour filling of drawn shapes
- printing of text anywhere on the diagram
- vertical and horizontal tabbing to give precise steps for cursor movement e.g. for placement of multiple lines of text.
- an erase feature which allows the cursor to act like an eraser to rub out portions of the diagram.

A proportional joystick or the cursor keys can be used to move the cursor, with or without leaving a trail in the currently selected drawing colour.

For movement by arrow keys the speed of the cursor can be varied faster or slower (this also affects the size of the eraser area). For joystick movement the speed depends on the stick displacement from the centre and the direction is determined from the joystick position.

The top of the screen contains a message area and a block of the currently selected colour. Colours can be changed by pressing 'C' and cycling through the available colours for the chosen mode.

Detailed descriptions of each program are as follows.

T-DRAW (Program 1)

This program displays a menu for obtaining a list of the function keys and command keys on the screen or on the printer, or a printed keystrip.

A final option initiates the chaining of the actual drawing program.

PROCEDURES (in alphabetic order)

PROCescape -
ensures that the printer is reset if Escape key is used

PROCinstructions -
displays the Menu and calls the output procedures as required

PROCkeystrip -
using Elite pitch to give 96 characters of printing width, this prints a suitably formatted function keystrip which also has a one-line summary of the main command keys

PROCprintend -
resets the printer and returns to screen output

PROCprintinit -
sets for printer output only and initialises the printer to give

emphasised printing

PROCprintinstr -
outputs instructions and control codes to the printer

PROCreadandprint -
gets next DATA line and then calls screen or printer output procedure

PROCscreeninstr -
outputs instructions and control codes to the screen; handles screen paging pauses

DATA

The same set of DATA is used for both screen and printer output of the instructions. Each DATA entry has a first character which indicates one of 6 types:-

S= screen line only
P= printer line only
B= data for both screen & printer
X= screen output control character
Y= end of page for screen output
W= printer control character

T - D R A W 2 (Program 2)

This program allows you to create drawings in Mode 0, 1 or 2; to save them to tape; and to print them.

The function keys can be used to call up a number of standard shapes (circles, squares, rectangles and lines).

Also the cursor can be moved freely, using arrow keys or joystick, to draw freehand shapes.

Shapes can be coloured-in by using a fill function.

The ability to write text anywhere on the drawing area allows charts or labelled diagrams to be prepared.

Mode 0 gives the sharpest drawings, but the screen is only in black & white. Mode 1 allows four screen colours and the printing is fairly sharp. Mode 2 allows use of all eight non-flashing colours but the screen graphics and printed diagrams are rather coarse.

PROCEDURES (in alphabetic order)

PROCcircle -
called from f1, calculates the radius, sets the starting position, then calls whichever of the 3 actual drawing routines is required. All circles have a centre which is set by the most recent use of f0, and a radius calculated from the current cursor position to the centre. Circle drawing can be interrupted by holding down the "Q" Key until drawing stops.

PROCcolour -
called by Key "C" and changes the graphics foreground colour to the next in the cycle for the current Mode; then calls PROCdisplaycol to update the colour panel.

PROCcross -
draws the cross which is the cursor for this program.

PROCdisplaycol -
defines the background colour to be the currently selected drawing colour; then defines a small text window in the top right corner of the screen and clears this to give a small panel of the chosen colour; the text window is then restored to its normal state for this program (ie approx 3/4 of the top two lines of the screen, which acts as a message area).

PROCdrawline -
called by f8 and draws a solid or dotted line (depending on Shift Key) from the last line-start position as set by f7. Several lines can be drawn from the same start position by just repetively moving the cursor and pressing f8.

PROCdump0/PROCdump1/PROCdump2 -
these are the variable parts of the printer dump routine which apply to Mode 0, 1 and 2 respectively.

PROCend -
called from CTRL-ESC Keys pressed together this carries out a few restoring activities to place the computer into a tidy state before leaving the program.

PROCerase -
called from PROCpointer when ERASE mode is on this turns the

cursor into an eraser which "rubs out" the parts of the drawing which it passes over. The size of the eraser depends on the current setting for cursor speed but can be varied by using the "+" and "-" keys. Either arrow keys or joystick can be used to move the cursor for this purpose.

PROCeraseonoff -
called from key "E" this turns the erase mode on and off.

PROCfill -
called from f9 this is a routine which provides some simple space-filling capability. Before pressing f9 the cursor should be placed at a suitable place in the middle of the area to be filled and the required colour selected. Then if we imagine a vertical line through the cursor position, the program scans vertically downwards until a colour which is different to that from where the cursor started is detected. This defines the lower boundary for filling. The program then searches both left and right from the imaginary reference line until a colour change occurs thus defining the sides of the area to be filled. A line is drawn between these two points and then the program moves upwards by the width of a line and repeats the searching and drawing. This continues until a colour change occurs in the vertical direction.
Because of the simplicity of the scanning algorithm, complex shapes may not get filled completely in one scan. In this case the cursor can be moved to the unfilled area and f9 again pressed.

PROChorizontal -
called from f5 this draws a horizontal line which runs the full width of the screen through the current cursor position.

PROCinit -
this initialises a multitude of items, including cancelling the normal cursor, setting the Escape key to give its Ascii code, getting the background colour, setting text and graphics windows, initialising a variety of coordinates and indicators, and drawing the cursor in its start position.

PROCinkey -
this is the main calling routine. It tests for all the various function and command keys being pressed, then calls the required PROCs.

PROCjoystick -
if joystick movement of the cursor has been requested then this routine is called to read the joystick position and calculate the required amount of vertical and horizontal movement.

PROClinecircle -
called from PROCcircle this draws the circumference of a circle.

PROClinerect -
called from PROCrectangle this draws the 4 sides of a rectangle whose size is determined by the most recent use of f7 which sets the bottom left corner and by the current cursor position which gives the top right corner.(This is similar to defining a text or graphics window).

PROCload -
called from CTRL-L this requests the name of the file to be loaded, and then uses PROCoscli to initiate a load command. Files should be loaded with the same Mode set as when they were saved, otherwise the effect is nonsensical.

PROCmodeselect -
presents a menu of mode options and waits for a valid selection.

PROCoscli -
calls the Command Line Interpreter in the Operating System.

PROCpatterncircle -
called from PROCcircle this draws a patterned circle formed by a series of radii drawn at 10 degree intervals.

PROCpointer -
determines whether the cursor has moved since last display, and draws the flashing cursor.

PROCprintscreen -
called from key "P" this calls the screen dump routine after making a density request to the user.

PROCrectangle -
called from f4 this initialises for rectangles and calls the required type of rectangle.

PROCsave -
called from CTRL-S this gets the desired filename (defaults to 'DRAWING"), and then uses PROCoscli to initiate a Save operation. The message area at the top of the screen is excluded from the saving.

PROCscreendump -
based on the screendumps given in 'Beeblet' of February and June

1983, this handles the printing of the drawing area of the screen. Printing is done sideways to give a large picture, and the routine has been tested on a Logitec FT-5000 which is reputed to be Epson-compatible. The printout can be interrupted by holding the "Q" key down until printing stops. (Note: for printers not set to give an automatic linefeed then a VDU1,10 should be included after the VDU1,13).

PROCsolidcircle -
called from PROCcircle this draws a solid circle of colour.

PROCsolidrect -
called from PROCrectangle this draws a solid rectangle of colour.

PROCsquare -
called from f2 this draws a solid square of colour, with the current cursor position determining the bottom left corner of the square. Interestingly, the square expands as f2 is held down, and will continue to expand even if interruptions occur in the holding of f2 so long as no colour change or cursor movement occurs.

PROCstartcircle -
called from f0 this notes the current cursor position to be the centre of a circle called from f1.

PROCstartline -
called from f7 this notes the cursor position to be the start of a line called from f8.

PROCstartrect -
called from f3 this notes the cursor position to be the bottom left corner of a rectangle called from f4.

PROCstephoriz -
called from key "H" this moves the cursor horizontally an amount as previously set (Key "S", see PROCstepset). The direction of stepping can be left (-ve step) or right (+ve step).

PROCstepset -
called from key "S" this requests either a vertical or a horizontal step amount.
PROCstepvert -
called from Key "V" this moves the cursor vertically an amount as previously set (Key "S", see PROCstepset). The direction of stepping can be up (+ve step) or down (-ve step).
PROCtext -

called from key "T" this routine inputs a text message to be printed on the drawing with the lower left corner of the first character being at the cursor.Reverse video effects can be obtained by drawing a solid rectangle of suitable size, and then changing to colour = black before using the Text command to print inside the rectangle.
PROCvertical -
called from f6 this draws a vertical line which runs the full height of the drawing screen through the current cursor position.

NB. THE MAIN DRAWING PROGRAM WILL BE PUBLISHED NEXT MONTH - it covers 5 pages. Copies are however available for anyone who cares to send in a tape or disc and stamped self addressed return packaging.

T-DRAW: C=Colour, T=Text, I=Invisible, P=Print, E=Erase, Q=Quit item, H/V=Tab, S=Set tabs

```
---------------------------------------------------------------------------------------
:  C I R C L E   :        :R E C T A N G L E:         :        :    L I N E    :        :
:----------------: ^      :-----------------:         :   :    :---------------:        :
:       :PATTERN : :      :        :        :         :   +    :       :       :  FILL  :
:   +   :--------: +-->   :        :--------:  --+--  :   :    :       :-------:  AREA  :
:  SET  : SOLID  :        :  SET   : SOLID  :         :   :    :  SET  :DOTTED :  WITH  :
:CENTRE :--------: SOLID  : START  :--------: HORIZ.  : VERT.  : START :-------: COLOUR :
:       :  LINE  : SQUARE : CORNER :  LINE  :  LINE   :  LINE  :       : LINE  :        :
---------------------------------------------------------------------------------------
```

```
 10 REM ** T-DRAW **
 20 REM
 30 REM ** Version 1.1 **
 40 REM  by W A Tillick
 50 REM
 60 ON ERROR PROCescape
 70 :
 80 REPEAT
 90   PROCinstructions
100   UNTIL A%=9
110 CHAIN"T-DRAW2"
120 END
130 :
140 DEFPROCinstructions
150 REM----------------
160 CLS
170 PRINTTAB(10,0);CHR$(141);CHR$(129)
    ;"T-DRAW"
180 PRINTTAB(10,1);CHR$(141);CHR$(129)
    ;"T-DRAW"
190 PRINTTAB(1,4);CHR$(131);"Do you wa
    nt instructions or keystrip?"
200 PRINTTAB(4,7);"1.  Instructions on
     Printer"
210 PRINTTAB(4,9);"2.  Instructions on
     Screen"
220 PRINTTAB(4,11);"3.  Print Function
     Keystrip"
230 PRINTTAB(4,13);"9.  Chain to Drawi
    ng program"
240 PRINTTAB(3,17);CHR$(131);"Enter ch
    oice ";
250 D$=STRING$(50," "):D$=""
260 REPEAT A%=VAL(GET$):UNTIL A%=1 OR
    A%=2 OR A%=3 OR A%=9:PRINT;A%
270 IF A%=1 OR A%=3 PROCprintinit ELSE
     CLS
280 IF A%<3 RESTORE 730:REPEAT PROCrea
      dandprint:UNTIL A$="9"
290 IF A%=3 PROCkeystrip
300 IF A%=1 OR A%=3 PROCprintend
310 ENDPROC
320 :
330 DEFPROCprintinit
340 REM-------------
350 *FX3,10
360 VDU27,64,27,69
370 ENDPROC
380 :
390 DEFPROCreadandprint
400 REM----------------
410 READ D$:A$=LEFT$(D$,1):B$=MID$(D$,
    2)
420 IF A%=1 PROCprintinstr
430 IF A%=2 PROCscreeninstr
440 ENDPROC
450 :
460 DEFPROCscreeninstr
470 REM---------------
480 IF A$="S" OR A$="B" PRINTB$:ENDPRO
    C
490 IF A$="X" PRINTCHR$(VAL(B$));:ENDP
    ROC
500 IF A$="Y" PRINT'CHR$(131);"Press a
    ny key to continue";:I%=VAL(GET$):
    CLS
510 ENDPROC
520 :
530 DEFPROCprintinstr
540 REM--------------
550 IF A$="P" OR A$="B" PRINTTAB(15);B
    $:ENDPROC
560 IF A$="W" PRINTCHR$(VAL(B$));
570 ENDPROC
580 :
590 DEFPROCprintend
600 REM------------
610 VDU27,64
620 *FX3,0
630 ENDPROC
640 :
650 REM DATA for instructions
660 REM----------------------
670 REM S = Screen only
680 REM P = Printer only
690 REM B = Both screen & printer
700 REM W = printer code
710 REM X = screen code
720 REM Y = pause (screen only)
730 DATA X129,X141,S      T-DRAW
740 DATA X129,X141,S      T-DRAW
750 DATA W14,PT-DRAW,B
760 DATA X130,B** MOVEMENT,P----------
    -,S,
770 DATA BArrow keys  = move cursor
780 DATA B  + Key     = speed cursor u
    p
790 DATA B  - Key     = slow cursor do
    wn,B
800 DATA BJoystick    = move cursor (i
    f selected)
810 DATA Y
820 DATA B,X130,B** COMMANDS,P--------
```

```
    ---,S
830 DATA BC = change colour
840 DATA BT = print text at cursor,B
850 DATA BI = select/deselect invisibl
    e mode
860 DATA BJ = select/deselect joystick
870 DATA BE = select/deselect erase mo
    de
880 DATA B,BP = print current drawing
890 DATA BQ = quit printing (hold unti
    l line end)
900 DATA B    or quit fill routine (ho
    ld)
910 DATA B    or quit circle immediate
    ly
920 DATA B,BS = set tab step (Vert. or
     Horiz.)
930 DATA BH = tab horizontally
940 DATA BV = tab vertically,Y
950 DATA B,X130,B** FUNCTION KEYS,P---
    --------------,S
960 DATA Bf0 = set centre of circle,B
970 DATA Bf1 = draw circle
980 DATA B     SHIFT-f1 = solid circle
990 DATA B     CTRL-f1 = patterned cir
    cle
1000 DATA B,Bf2 = solid square (expandi
    ng),B
1010 DATA Bf3 = start rectangle
1020 DATA Bf4 = draw rectangle
1030 DATA B     SHIFT-f4 = solid rectan
    gle
1040 DATA B,Bf5 = horizontal line
1050 DATA Bf6 = vertical line,B,
1060 DATA Bf7 = start a line
1070 DATA Bf8 = draw a line
1080 DATA B     SHIFT-f8 = dotted line
1090 DATA B,Bf9 = fill a shape,Y
1100 DATA B,X130,B** SPECIAL COMMANDS,P
    ------------------,S
1110 DATA BCTRL-S = SAVE current drawin
    g
1120 DATA BCTRL-L = LOAD a saved drawin
    g,B
1130 DATA BCTRL-DEL = Clear screen,B
1140 DATA BCTRL-ESC = End program,Y,W12
1150 DATA 99999
1160 :
1170 DEFPROCescape
1180 REM----------
```

```
1190 VDU1,27,1,64
1200 *FX3,0
1210 REPORT:PRINT" at ";ERL
1220 END
1230 :
1240 DEFPROCkeystrip
1250 REM------------
1260 VDU27,80,0
1270 PRINTTAB(2)"T-DRAW:  C=Colour, T=T
     ext, I=Invisible, P=Print, E=Erase
     , Q=Quit item, H/V=Tab, S=Set tabs
     "
1280 PRINTTAB(1)"______________________
     __________________________________
     __________________________________
     _"
1290 PRINTTAB(1)"¦   C I R C L E   ¦
          ¦R E C T A N G L E¦        ¦
            ¦    L I N E    ¦
     ¦"
1300 PRINTTAB(1)"¦-----------------¦ ^
          ¦-----------------¦        ¦
        ¦   ¦-----------------¦
     ¦"
1310 PRINTTAB(1)"¦         ¦ PATTERN¦ ¦
          ¦        ¦         ¦        ¦
        +   ¦        ¦         ¦ FILL
     ¦"
1320 PRINTTAB(1)"¦   +    ¦--------¦ +-
     -)   ¦        ¦--------¦ --+--  ¦
        ¦   ¦        ¦--------¦ AREA
     ¦"
1330 PRINTTAB(1)"¦  SET   ¦ SOLID  ¦
          ¦  SET   ¦ SOLID  ¦        ¦
        ¦   ¦  SET   ¦ DOTTED ¦  WITH
     ¦"
1340 PRINTTAB(1)"¦ CENTRE ¦--------¦ SO
     LID  ¦ START  ¦--------¦ HORIZ. ¦
      VERT. ¦ START  ¦--------¦ COLOUR
     ¦"
1350 PRINTTAB(1)"¦         ¦  LINE  ¦ SQ
     UARE ¦ CORNER ¦  LINE  ¦  LINE  ¦
      LINE  ¦        ¦  LINE  ¦
     ¦"
1360 PRINTTAB(1)"----------------------
     ----------------------------------
     ----------------------------------
     _"
1370 ENDPROC
```

STACKING COINS

Anton E.

This is the first program I have written for some months, but my wife is recovering from a prolonged bout of morning sickness, and is willing to suffer competition again. The program is a formal model of a problem in mechanics: how far out can you lean a stack of coins? Answer: as far sideways as you want, but it can take an awful lot of coins to do that.

For those with a mathematical bent, the overhang can reach a maximum of about 0.29+0.5*LN(x) coin widths, where x is the number of coins. I wont give you the exact relationship, partly because it would remove some interest from the problem, but mainly because there is no exact formula: There is a descriptive series possible, with a term for each and every coin. By starting with the simplest case,(2 coins), the stability limit is easily found. Then try 3 coins. And so on. If you dont find the relationship, you haven't much chance of reaching the challenge target.

Mind you, I haven't reached the target myself, but I think I could. I have found that the lazy way to get a tall stack (holding the N key down) is a rapid way to doom. Stability is checked from top coin downwards, and slovenly stacking generates a "burnout" at the first point a cumulative center of gravity finds itself perched on fresh air.

The program is well structured, but in vile defiance of Tim R., I have indulged in no-nos here and there. A double exit from PROCnewcoin, for instance. The "game" should be non addictive, but frustrating, and readily mastered by the sort of supercilious little brat that makes me hate Rubic cubes and resort to words like supercilious and brat.

And a final warning! Dont leave out PROCtidyup or the cursor keys will leave you cursing whenever you escape from the program. Have fun.

EDUCATIONAL SOFTWARE - ELECTRONS

I have been asked several times where "good" "educational" software for the Electron can be obtained. Try:

1. Writing to our advertisers and other suppliers for their catalogue. Try and be specific about your requirements.
2. Getting a copy of "Software Index" - available in NZ, published by IPC Magazines Ltd., Lavington House,25 Lavington St.,London SE1 0PF: 2·Pounds.
3. Sending specific details of what you want to Beeblet.

****

L.STACKIT

****

```
  1 REM *** Coin stacking ***
  2 REM Game models gravity effects
  3 REM Anton E.,Jul'85.
  4
 10 MODE1
 20 ON ERROR GOTO 130
 30 PROCsetup
 40 REPEAT
 50   PROCfirsttwo
 60   REPEAT
 70     PROCshift
 80     PROCdisplay
 90     PROCstable
100     UNTIL flag%>0
110   PROCfailure
120   UNTIL bored
130 PROCtidyup
140 END
150
160 DEF PROCsetup
170 DIM penny(42)
180 record=0:side%=1:bored=0
190 @%=&20608
200 un$ = STRING$(20,"Unstable! ")
210 *FX4,1
220 VDU26:CLS
230 PRINT'" The aim of the game is to
    stack coins."
240 PRINT'"Lean them out as far as gra
    vity allows."
250 PRINT'TAB(11)"The challenge:"
260 PRINT'TAB(6)"A top coin overhang o
    f"
270 PRINT'" two coin widths using only
     32 coins."
280 PRINT''"Use the cursor keys to mov
    e a hammer,"
290 PRINT'"and hit using a digit, from
     number"
300 PRINT'"1 (powerful) to 6 (weak). T
    he overhang"
310 PRINT'"of each coin (above the bot
    tom one),"
320 PRINT'"and its position relative t
    o the one"
330 PRINT'"just beneath it are both gi
    ven."
340 PRINT'"N gives a new coin, randoml
    y placed!"
350 PRINT''"      Space bar to continu
    e."
360 g%=GET
370 VDU28,0,31,20,0
380 ENDPROC
390
400 DEF PROCfirsttwo
410 CLG
420 MOVE 700,0:DRAW 1280,0
430 penny(0)=2:REM Stack no. kept here
440 level%=2
450 MOVE 700,4:DRAW 1280,4
460 penny(1)=0:PROCcoin(1)
470 penny(2)=.1:PROCcoin(2)
480 ENDPROC
490
500 DEF PROCshift
510 PRINTTAB(5,12)"Options:"
520 PRINT'" Cursor Keys"
530 PRINT'" Digit ( 1 to 6 )"
540 PRINT'" N = new coin"
550 REPEAT
560   PROChammer(level%,side%)
570   PRINTTAB(1,22)" Coin ";level%
580   PRINT'"Overhang ";penny(level%)
590   PRINT'"Relative ";penny(level%)-
      penny(level%-1)
600   g%=GET
610   PROChammer(level%,side%)
620   IF g%=136 THEN side%=-1
630   IF g%=137 THEN side%=1
640   IF g%=138 THEN level%=level%-1
650   IF level%<2 THEN level%=penny(0)
      :VDU7
660   IF g%=139 THEN level%=level%+1
670   IF level%>penny(0) THEN level%=2
      :VDU7
680   g%= (g%OR32)
690   UNTIL (g%>48 AND g%<55) OR (g%=1
    10)
 700 IF g%=110 THEN PROCnewcoin ELSE PR
     OCnudge
 710 ENDPROC
 720
 730 DEF PROCdisplay
 740 FOR I% = 1 TO penny(0)
 750   PROCcoin(I%)
 760   NEXT
 770 hangout = penny(penny(0))
 780 CLS:PRINT '" Top coin position:"
 790 PRINT''"Overhang is "; hangout
 800 PRINT''"Record = ";record
 810 ENDPROC
 820
 830 DEF PROCstable
 840 sum = 0:flag% = 0:scan% = penny(0)
 850 REPEAT
 860   sum = sum + penny(scan%)
 870   scan% = scan%-1
 880   grav = sum/(penny(0)-scan%)
 890   IF ABS(grav-penny(scan%)) > .5 T
       HEN flag%=scan%
 900   UNTIL flag%>0 OR scan%=1
 910 IF flag%=0 THEN PROCokay
 920 ENDPROC
 930
 940 DEF PROCfailure
 950 VDU7
 960 PRINTTAB(1,5) un$
 970 PLOT70,1000+100*grav,4+24*flag%
 980 FOR I%=1 TO 250
 990   PLOT0,RND(5)-3-I%,4:PLOT2,I%,0
1000   NEXT
1010 PRINT'''"  Why not join the"
1020 PRINT'" Ministry of Works!"
1030 PRINT''"PRESS B if bored"
1040 *FX15,0
1050 PRINT'"else the space bar."
1060 g%=GET
1070 IF (g%OR32)=98 THEN bored=1
1080 ENDPROC
1090
1100 DEF PROCcoin(K%)
1110 GCOL 0,1+K%MOD3
1120 MOVE 934+100*penny(K%),24*K%-16
1130 PLOT 0,0,20:PLOT83,16,-20
1140 PLOT83,0,20:PLOT81,100,-20
1150 PLOT81,0,20:PLOT83,16,-20
1160 PLOT83,0,20
1170 ENDPROC
1180
1190 DEF PROCnewcoin
1200 IF penny(0)>42 THEN VDU7:ENDPROC
1210 penny(0)=penny(0)+1
1220 level%=penny(0)
1230 penny(penny(0))=penny(penny(0)-1)+
     (RND(21)-11)/20
1240 PROCcoin(penny(0))
1250 ENDPROC
1260
1270 DEF PROCokay
1280 PRINT'" System is stable"
1290 IF ABS(hangout) > ABS(record) THEN
      PROCfame
1300 ENDPROC
1310
1320 DEF PROCfame
1330 CLS
1340 PRINT''"NEW RECORD OVERHANG"
1350 PRINT''"Beats old record by"
1360 PRINT'record-hangout;" widths"
1370 record = hangout
1380 PRINT'" New record"
1390 PRINT'"  is..  "; record
1400 ENDPROC
1410
1420 DEF PROCnudge
1430 offset=1
1440 REPEAT
1450   offset=offset/10
1460   g%=g%-1
1470   UNTIL g%<49
1480 FOR I%=level% TO penny(0)
1490   penny(I%) = penny(I%)-offset*sid
       e%
1500   NEXT
1510 ENDPROC
1520
1530 DEF PROChammer(A%,B%)
1540 MOVE 1000+penny(A%)*100+B%*60,24*A
     %-4
1550 PLOT2,40*B%,0
1560 PLOT0,0,4:PLOT2,-40*B%,0
1570 ENDPROC
1580
1590 DEF PROCtidyup
1600 @%=&A0A
1610 *FX4,0
1620 VDU26
1630 GCOL 0,3
1640 ENDPROC
```

ADVENTURE GAMES

Bryan H.

This has all been one big adventure! Not the sort you think, playing a game, but trying to understand how an adventure game program works. Enter the Evil Genius G.C.F. with Program No 2 in which he has incorporated all the features needed to run complicated adventure games. Holding aloft a 12 page summary of how it all works he chants the inventor's inevitable assurance "It is really quite simple...........".

G.C.F. has already written four articles on the subject, spent three months solid designing the beastly thing and more writing the program. You are expected to learn how to use it in five minutes, plan and write your first adventure in the next five minutes, and give a balanced assessment of its potential straight after. Result is confusion. This article sets out to be a introduction. Hopefully after reading through you will know what is available for the adventure writer and how to put in your adventure. Only relatively simple examples will be used. Some of the complications will turn up this time; adventure addicts will have to wait until later for the curly bits.

HOW IT WORKS

The program like its little brother VSAS (Beeblet APRIL 1985) consists of a main program and a heap of data statements at the end. The program is technically an interpreter. That is, it reads the data statements, interprets them as commands, text etc. and then obeys the commands.

The effect as far as the user is concerned is for descriptions to be printed on the screen and a menu of choices presented. After you chose an alternative there may be some consequences (quite often nasty in an G.C.F. adventure) and perhaps a move to another part of the adventure.

For the writer of an adventure the data statements can be considered as a computer language. You don't have to worry how the interpreter works, only how to use the language to get your adventure working. Later in this article I will and work through the statements in detail.

FRAMES

Building blocks are called frames, and each one describes something in the adventure. Some are places some are objects eg. boxes, lamps. Some are creatures, dragons perhaps. As the adventure proceeds the program writes out selected descriptions and presents the user with alternative courses of action. Depending on the action selected, the player may pick up an object, move to another frame, or perhaps be destroyed by an explosion. It is all up to the adventure writer.

Each object described in a frame can be inside another. You specify the initial position and the program keeps a list of which frame each frame is enclosed in, and the number held in each frame. See the diagram for an example. There is no restriction built in to the program, anything can go in anything, you can put the whole adventure inside the lamp if you have the urge. Initial positions are shown at the begining of a frame eg. #12/23 means #12 will first be found inside #23.

Frame numbers #0 and #1 are reserved, otherwise frames can have any number from #2 to #255. Frame #0 is the player's frame. Options specified under the player are checked each step of the adventure eg. a command to list all objects (frame title) in the room can be included in the player's commands - described if it is not invisible of course (invisibility is indicated by flag 0 being set on). The program has a specific order of presentation. At each step of the adventure the order of action is:

1 Process the player's descriptions.

2 Process the current frame's descriptions.

3 Process the current frame's options.

4 Process the player's options.

5 Process current frame's follow-ups.

6 Process player's follow-ups.

The current frame is the frame the player is currently in. Options for the player is to allow the user to choose an action eg move to another room, examine or pick up an object. Once picked up, an object is enclosed in the player's frame, or, more descriptively, carried by the player.

DATA STATEMENTS

At the beginning of the Data are a series of statements starting with the symbol "~". These are commonly-used phrases, but in the interests of RSI and RAM, are written out once, then called by number: you can see them used later on in the text by eg "~1" The program adds spaces before and after these phrases (except where followed by punctuation):

```
   5000 DATA ~1,tree covered hillside
   5180 DATA $,You are on a#21beside a~1.
```

Comes out as:

```
   You are on a flat grassy lawn beside a tree covered hillside.
```

In line 5180 the #21 also represents a phrase, but comes from a different place - it is the title of frame 21.

After all the ~ phrases come the main body of data - the frame descriptions. Each frame begins with # and a number, which is the frame's I.D. Each frame must have a title.

Within the frame data lines begin with $ % or & and are specified in that order. Each type of data has 3 parts to it, each separated by a comma:

1. a symbol ($%&) to indicate the type, plus any qualifying conditions.
   - $ indicates a description follows,
   - % indicates an option for the user to choose
   - & indicates a followup after an option has been chosen.
2. a phrase to be printed out (optional)
3. Commands etc set, follow-ups to be actioned. (optional)

Let's work through frame 24:

```
5470 DATA #24,narrow tunnel
5480 DATA $e12<>0,The#24is dark and you can only see some light
behind_ noises of rats ahead.
5490 DATA $e12=0,You can see the tunnel bending to the left and
ascending_ it looks well used.
5500 DATA %,Go on up the tunnel.,25
5510 DATA %,Stay and fight.,ST2
5520 DATA %,Go back.,ST2
5530 DATA &t2,You are captured.~2,AX
```

line 5470 has: "#" at the very beginning, meaning "this is a new frame being started."
"24" is the I.D. Followed by a comma as a separator;
"narrow tunnel" is the title.

5480 begins with $ to indicate that this is a line of text to be

printed out;
e12<>0 is a condition which has to be fulfilled before that part of the frame description can be printed out. In this adventure, frame 12 is a lamp, and is not in frame 0 which is the player. i.e. if the player does not have the lamp, print out the following:
"The narrow tunnel is dark and you can only see some light behind, noises of rats ahead."
This doesn't exactly look like what is in the data statement, but it is what will end up on the screen.
5490 you can probably figure out that if the player has a lamp, he will get this message instead.

Here there follow three options for the user to choose from. To show this they all begin with %. Like $, % can also have conditions. Then comes the text for the user to read, and the third part is what to do if the user selects that option.
5500 If you chose this option, the player would be moved to frame 25.
5510, 5520 either of these options instead of moving to another frame, sets a flag for the & line to deal with.

5530 & - this is a followup.
t2 - a condition of actioning the followup is that flag t2 in this frame be set - in which case: You are captured, phrase ~2 (no relation to flag 2) is also printed out, and the command AX is actioned. AX means End Of Game, End Of Program, the AXe Has Fallen - or more prosaically, Adventure eXit.

FRAME STATUS FLAGS

Each frame can have attributes decided by the adventure writer. Some are variables. Some are flags, that is, can be expressed in terms of set on and set off. See the list below.
Flag 0 is used by the program as a test of invisibility. on = invisible, off = visible. Flags 1-16 can be used for anything you like. In frame #24 flag 2 was used to condition the followup to options chosen. Other flag uses can include lit/unlit for lamps, open/shut for doors and boxes. Flags can be set on at the start of the adventure and can be changed by commands.

OTHER COMMANDS

What commands are there? Lots!
A few of the 38:

AX	Adventure eXit
CTx	Clear This frame's flag x (set off)
DDx	Display Description of frame x
DVx	Display list of Visible items in frame x

GD	Generate options to Drop items
GEx	Generate options to Examine item
GTx	Generate options to Take items in x
MEx/y	Move to Enclose x into frame y
STx	Set This frame's flag x on
VRx/y	Value x assigned to Register y

VARIABLES

There are 11 "SAS system variables"

cf	Current Frame number
pf	Previous Frame number
nt	Number of Turns so far (used in #23 for instance)

.... 100 registers........r0 to r99

PLANNING ADVENTURES

Now to look at the adventure itself. Yes, I followed GCF's advice and drew a diagram. Yes, I carefully worked out what flags would be used where and for what. No, I did not get the descriptions quite right:

```
  5320 DATA #23,circular room
  5330 DATA $,You are in a#23
  5340 DATA $,^A#24is on the right
  5350 DATA $,^An ornate golden arch is on the left with a#26lea
ding away.
  5360 DATA $,^Ten dwarfs are approaching down the#26_ they don'
t look friendly.
  5370DATA $NOTt3,^You have time to pick up two objects before t
hey arrive,ST3VR(nt+2)/1
  5380 DATA %,Go into tunnel on right,24
  5390 DATA %,Stay and fight.
  5400 DATA %,Go back,22
  5410 DATA &r1<nt,You are captured~2.,AX
```

The room #23 descriptions for instance work fine if you are going my assumed way through the adventure. Left and right only work if you are facing the 'correct' way as you enter - there are three ways into the room... You could have three sets of descriptions, each dependent on which room you were in last (using variable pf). This is hard work and unnecessary unless you are trying to test the player's memory. Descriptions are better related to north and south if possible. Forward and back also depend on which way you are going - oops again! (Note ^ starts a new line).

Fumble number two was not making the dwarves a frame. They could easily be induced to follow the player around by moving them each turn to the player's previous frame. Setting limits on their movements, such as staying underground, would also be possible. As it stands, there is an inexhaustible supply: every time you arrive at #23 10 more pesky dwarves appear.

By now you may be getting a general impression of how it works. I wanted to include a whole adventure for you to work on but the editor is threatening me with a club! Ah well, next issue perhaps.

EDITORIAL ASIDE

SAS is now available (version 0.95 or later) with an example adventure of some 40-ish rooms
plus the paperwork listing commands, syntax...
From the User Group P.O.Box 9592 Wellington,
To any member who sends in a disc (or tape), return postage, plus $1 for the photocopying. And quotes their membership number of course.
FEEDBACK is also essential, otherwise PAS will hardly get PASt dreamland. (Remember March 'Beeblet'?)

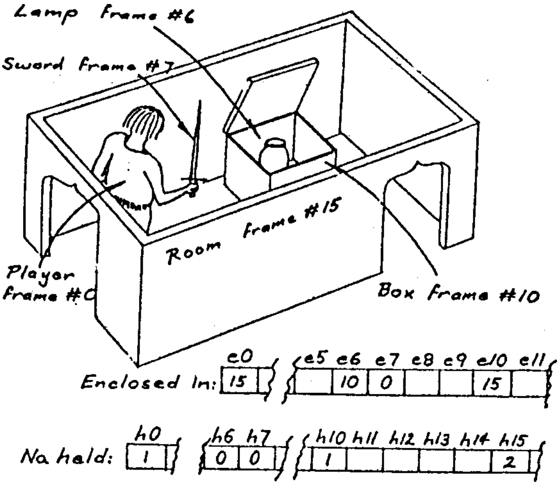
```
 **********************************

           L.ADDER

 **********************************

  10 REM::ADDER:: N.E.BAKER 2/84
  20 @%=10
  30 MODE7
  40 DIM N(300)
  50 total=0
  60 X=0
  70 PRINT'"Enter 0 when Total required
     "
  80 PRINT
  90 REPEAT
 100   X=X+1
 110   PRINTX".";
 120   INPUT"        "value
 130   N(X)=value
 140   total=total+value
 150   UNTIL value=0
 160 PRINT STRING$(28,"_")
 170 @%=&20209
 180 PRINT"TOTAL","     "total
 190 REPEAT
 200   PRINT'"press RETURN for readout"
 210   PRINT"press Space Bar for correc
       tion"'
 220   Y=0:total=0
 240   IF GET=32  PROCcorrect
 245   REPEAT
 250     Y=Y+1
 260     @%=10
 270     PRINTY".";
 280     @%=&20209
 290     PRINTN(Y)
 300     UNTIL Y=X-1
 310   PRINT STRING$(28,"_")
 320   FOR M=1TOY
 330     total=total+N(M)
 340     NEXT
 350   PRINT"TOTAL","  "total
 360   UNTILFALSE
 370
 380 DEF PROCcorrect
 385 REPEAT
 390   INPUT "Enter no. of faulty entry
        ",F
 400   IF F)X-1 SOUND1,-15,222,5
 405   UNTILF(X
 410 INPUT "Enter correct value ",corr
 420 N(F)=corr
 430 ENDPROC
```

```
You are carrying a sword. You are in a room with exits south and
east. There is an open box with a lamp.
  A - south
  B - east
  C - examine lamp
  D - examine box
  E - take lamp
  F - drop sword                  Option ?
```



Part of the listing would approximate this:

```
  DATA #15,room
  DATA $,You are in a#15with exits south and east. There is an o
pen#10
  DATA $e6=7,with a#6
  DATA $,.
  DATA %,south,14
  DATA %,east,16
  DATA %e6=7,examine#6,DD6
  DATA %,examine#10,DD10
  DATA %h0<3,take#6,ME6/0
  DATA %e7=0,drop#7,ME7/15
```

ELOPE

WITH THOSE ENVELOPES

Hataitai Workshop

Ever heard a neat sound in some program sitting in your machine, and wondering what the envelope was so you could use it in a program of your own? But can't get at a listing to find out?

Why not look it up in the pages of your computer? Addresses &8C0 to &8FF are where envelopes 1 to 4 are stored. While you are not using &900-&9BF for either RS423 or cassette output, envelopes 5-16 can be stored in here.

The program below looks through this part of memory and prints out the envelope definitions in a more recogniseable form. You just have to copy the numbers off the screen and insert in your next program to rival Space Commander Yehudi Menuhin!!

Not only does this program print out the envelope definition, it also makes a sound using that envelope, then waits for you to press a key to demonstrate the next envelope for you.

```
   10 REM recover envelopes
   20 REM next 4 lines set up use of OSWORD 5
   30 DIM block% 4
   40 X%=block% MOD 256
   50 Y%=block% DIV 256
   60 A%=&05
   70 FOR I%=1 TO 16
   80   PRINT "ENV.";I%;
   90   FOR J%=2 TO 14
  100     !block% = &08C0 + 16*(I%-1) + (J%-2)
  110     CALL &FFF1
  120     N% = block%?4
  130     IF NOT (J%=2 OR J%=6 OR J%=7 OR J%=8) AND
 N% >127 THEN N% = N%-256
  140     PRINT",";N%;
  150     NEXT J%
  160   SOUND1,I%,100,25:SOUND1,0,0,0:D%=GET
  170   PRINT
  180   NEXT I%
  190 END
```

Some envelopes may suit a different approach to discover their richness. Line 160 could be changed to:

`160 PROCsound:D% = GET`   and therefore add:

↓↓

```
  200 DEF PROCsound
  210 FOR K%=1 TO 8
  220    SOUND1,I%,100+K%*2,1
  230    NEXT K%
  240 FOR K%=8 TO 1 STEP-1
  250    SOUND1,I%,100+K%*2,1
  260    NEXT K%
  270 SOUND1,0,0,0
  280 ENDPROC
```

And while we're looking at sounds, how about this rhythm to try out. Note that line 60 clears the keyboard buffer. That is so you can escape the program by pressing any key:
line 140 ADVAL -1 tests this

```
   10 REM developed initially on Quicksilva 's Drum
Kit
   20 ENVELOPE1,1,-2,0,0,1,0,0,127,0,0,-10,80,0:REM
  "snare"
   30  ENVELOPE2,1,0,0,0,0,0,0,127,0,0,-20,60,0:REM
 "sticks"
   40 ENVELOPE3,2,0,80,-4,1,20,2,127,0,46,-3,70,0:R
EM  "electro"
   50 ENVELOPE4,1,-15,30,-3,2,1,10,127,0,0,-4,120,0
:REM  "bass drum"
   60 *FX15,1
   70 REPEAT
   80    RESTORE
   90    FOR I%=1 TO 16
  100      READ A%,B%,C%,D%
  110      SOUND0,1*A%,6,5:SOUND1,4*B%,30,5:SOUND2,3
*C%,32,5:SOUND3,2*D%,190,5
  120      SOUND 0,0,6,0:SOUND2,0,6,0:SOUND1,0,6,0:S
OUND3,0,6,0
  130      NEXT I%
  140    UNTIL ADVAL(-1)
  150 DATA 0,0,0,0,1,1,1,0,1,0,0,0,0,0,0,1,1,0,0,0,
1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0
,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0
```

****

THIS SPACE AVAILABLE FOR ADVERTISEMENTS

↑↑

LOOK! NO FILES!

A SIMPLE DATABASE GROWS I

Well, thought the Plunket committee, we should mail out the AGM notices, rather than deliver them by hand to 200-plus families in what must be one of the hilliest suburbs in Wellington. There is also this continuing problem of figuring out whose child needs an appointment for the 3-year-old doctor clinics. Not to mention the fact that with no Plunket nurse for the past year, our "records" were sadly out of date.

So why put "records" on a computer in the first place? Pointless without access to a printer really, so let's assume you can get hold of one. Now, you can update a phoning list without having to retype the lot, or produce lists of people under selected criteria for a club or group.

There are two beebs in our Plunket committee, and the OSI has recently acquired the Play Centre address list.... (I bethought me of our User Group database which has <u>everything</u> you put on the membership form in it; of how I spend hours abusing one of the several programs and files attached, to do anything more exotic than address labels, and quailed...) Something simpler perhaps, that didn't need me to learn about disc/tape files, Random Access... After all, there are only 6 lines for each sticky address label. One can't do much with them, especially as one line is actually the gap between labels; leave out one more line to cope with printer alignment - that would leave only 4 lines to muck up!

Something along the lines of:

```
READ name$, street_address$
PRINT"Mrs ";name$
PRINT street_address$
PRINT"Hataitai"
PRINT"Wellington"
PRINT
PRINT
END
DATA Mary Adams,142 Tiriti St    ....
```

should do nicely, shouldn't it?

Good idea, we enthused. So Keran and I set about updating our paper records, then arranged to get together to type in the data, to churn out address labels. Meanwhile, I mused upon the lists we used to have of families in each street (so many streets per committee member), the 3-year-old clinic... and came up with a format for the data.

1. Split people's names into first name or initial, and surname: so we could have address labels for "Mary Adams", and also alphabetical lists looking like "Adams, Mary"
   So we have surname$ and first_name$
2. Separate the street number from the street name.
   To produce lists of each street without having to get the program to decide what was number and what was street name
   Ended up with street_no$ and street$
   Why make the street number a string rather than a number (eg street_no%)? Well *you* try getting the computer to treat "19A" as a number, or "102 bottom flat" as a number. It falls over like a runny jelly!
3. Add in the phone number. phone$
   Make it a string, not a number, so you can include the hyphen (888-111) or cope with a party line (1101K)
4. Date of Birth of child has to be put in a consistent format so we could do "arithmetic" with it. I decided to make a decimal number: the last two numbers of the year, decimal point, month of birth. So Jan 1983 became 83.01, and October 1981 became 81.10 (or 81.1 would do). Ended up with dob$ (dob = Date Of Birth) - another number being referred to as a string! Why?? Because not everyone had only one child under 5, and the simplest way I could think of to get the computer to keep reading data as dob$ and not the next person's name, was by putting a special symbol at the end of the last dob. So one mother could have children with 81.01,83.11,85.01@ dobs. It is easy enough to strip off the "@" and then treat dob$ as a number.
   P.S. watch the commas and decimal points (full stops) don't get muddled.

So now we have a format for typing in all the DATA we had about each family under Plunket care:

```
DATA surname$,first_name$,street_no$,street$,phone$,dob$@
```

What if we didn't have a phone number, or some other piece of information? Simple! Just leave out, BUT KEEP IN THE COMMA so the computer continues in the right place. eg

```
DATA Jones,Mary,17,Tiriti St,,84.06@
```

THIS SPACE FOR PRODUCTIVE HIRE

(Q. What do you call a cross between a budding adventure system designer and a loquacious beanstalk?
A. A gabbling flower.)

↓↓

Came the day for Keran and I to type in the data.

```
AUTO 1000
```

and away we went. Finished everybody, ESCAPE

```
AUTO 10
   10 REPEAT
   20   READ surname$,first_name$,street_no$,street$,phone$
   30   REPEAT
   40     READ dob$
   50     UNTIL RIGHT$(dob$,1) = "?"
   60   IF surname$ <>"END" THEN PROClabel
   70   UNTIL surname$ = "END"
   80 END
   90 DEF PROClabel
  100 PRINTfirst_name$;" ";surname$
  110 PRINTstreet_no$;" ";street$
  120 PRINT"Hataitai"
  130 PRINT"Wellington"
  140 PRINT
  150 PRINT
  160 ENDPROC
```

Looks nice and straightforward, doesn't it. Read in all the data about a family, and use what we want of it. Did you notice that the computer READs until it gets to someone called "END"? A "dummy record" is put at the end of the data to give the computer some idea of where to stop!

How will we get everybody into alphabetical order of surname for a different list? Easy! Type people in in alphabetical order!! Saves writing a PROC to sort them into order (which is beyond me still).

Well, there are the labels for the AGM. Next time we'll look at resulting full-blown program, and I'll even include the listing!! Then you can flog the PROCs for your own uses.

↓↓

MICRO COVERS

Designed to fit the BBC micro . Attractive opaque reinforced plastic, $5-50 posted. To Iain, Box 26078, Epsom, Auckland.

CASSETTE TAPES

C10s, boxed, leaderless - available from the UG in multiples of 20 : $26-00 plus $2-00 p&p.

FLOPPY DISCS

3M brand, 5 1/4" DSDD discs. Available from the UG in cardboard boxes of ten: $55-00, which includes p&p.

↑↑

SAVING SCREENS

Hataitai Workshop

Sometimes a screen picture is worth keeping for posterity - either on paper or for future redisplaying on screen.

1. "SCREEN DUMPS" TO PRINTER.
If you haven't already, give your bank manager a hernia and buy a printer. Plug it in and tuck some little machine code routine somewhere in your computer's memory (or spend up big to buy some ROM like GDUMP). When you have a screen you want to copy, call this little routine to print an exact copy of the screen on to paper.

2. SAVE ON TAPE OR DISC
You can use either tape or disc to save details of your screen, then load them back onto the screen another day.
Whereas Basic programs are SAVEd for future use, a screen display is *SAVEd.
To do this you also have to specify which area of memory is to be saved, and this varies between modes, though not between Electron and BBC.

START ADDRESSES:

Modes 0, 1, 2 all start at	&3000
Mode 3 at	&4000
Modes 4, 5	&5800
Mode 6	&6000
Mode 7	&7C00

(These numbers assume that the screen has not scrolled since the last Mode change or CLS)

To save a Mode 1 screen for instance, type in:

```
*SAVE screenname FFFF3000 FFFF7FFF
```

(If you don't expect to be using a co-6502, you could type in:

```
*SAVE screenname 3000 7FFF
```

To get it back you need to be in the same mode you saved from:

```
MODE 1
*LOAD screenname
```

It may pay to start the filename with a number to represent the mode eg "1name" because...

Eventually you will have several screens that you want to bring up as a non-interactive display (ie no-one touches it). You can then write a little program that will load each screen in sequence with a pause after each one. If the files all begin with the mode number, the program can choose what mode to set automatically, and will look like this:

```
   10 REM screen dump menu program
   20 REPEAT
   30   RESTORE
   40   FOR I%=1TO5
   50     READ file_name$
   60     MODE VAL(file_name$)
   70     OSCLI("LOAD " + file_name$)
   80     wait = INKEY(5000)
   90     NEXT I%
  100   UNTIL FALSE
  110 END
  120
  130 DATA 1ATRIK,3ATRIKB,2RHODIN,1BATROK,1LOTFUN
```

The DATA words will be the names of your files on disc/tape. (If you have a BBC with Basic I refer to Beeblet Dec'83 for a way to use OSCLI)

The screens do not all have to be from Basic programs - this has been used with screens generated from a mixture of languages such as LOGO, BASIC, machine code... Some day when you do buy a printer, you can change line 80 to read:

```
  80 PROCprinter_dump
```

or whatever it will be.

L.MACSIM

```
 10 REM MACINTOSH SIMULATION
 20 REM BY TIM RYAN
 30 MODE 0
 40 FOR T=0 TO 1000 STEP 4
 50   REMMOVE 0,T
 60   MOVE -(T MOD 8 DIV 2),T
 70   PLOT 21,1278,T
 80   NEXT
 90 VDU 24,250;200;950;800;
100 CLG
110 GCOL 0,129
120 VDU 24,270;220;970;820;
130 CLG
140 GCOL 0,0
150 FOR T=780 TO 820 STEP 8
160   MOVE 270,T:PLOT 17,250,0:MOVE 72
      2,T:PLOT 17,250,0
170   NEXT
180 MOVE 270,772:PLOT 17,700,0:MOVE 27
    0,820:PLOT 17,700,0
190 VDU 5
200 MOVE 560,810:PRINT "Graphics"
210 GOTO 210
```

HALL OF FAME

PUBLISHER:	GAME	HISCORE	SCORER
Aardvark	Frak	1,026,820	Andrew P.
	Monsters	175,110	Geof P.
	Zalaga	215,180	Peter J.
		52,090*	Nick C.
Acorn	Arcadians	34,740	Rasik R.
	Aviator	5,610	Joanne C's Dad
	Countdown	246	Andrew
	Meteors	46,200	Andrew P.
	Monsters	149,730	Rupert T.
	Planetoid	413,875	Peter J.
	Rocket Raid	140,777	Mark R.
	Snapper	250,000	Susan
	Starship Command	9,495	Robert N.
	Super Invader	59,250	Kiran R.
A&F	Cylon Attack	157,970	Aaron
	Chuckie Egg	319,450*	Greg C.
Alligat.	Bug Blaster	37,642	Steve W.
	Guardian	7,900	Toni I.
Amcom	Fortress	89,900	Andrew P.
Bugbyte	Twin Kingdom	385	A. Beeston
Comp.Con	Android Attack	236,195	Peter J.
Dr Soft	'747'	205	Steve W.
Firebird	Bird Strike	11,340	David G.
		14,980*	Patrick C.
FG Soft.	Mutant Defender	6,900	Steve W.
	Lords of Time	925/1000	Steve W/Kerry K.
	Snowball	900/1000	Steve W./Dianne
Prog. Pow	Felix in Factory	13,150	Duncan's Flatmate
	Frenzy	915,468	Dianne T.
	Killer Gorilla	91,000	Susan
	Laser Command	109,225	P. Leadley
	Moon Raider	112,350	P.Leadley
Sup.Soft	Q*Bert	43,825	Peter J.
	Mr Wiz	21,050*	Chris M.
	Roadrunner	118,900	Duncan's Flatmate
?	Dr Who	35,240	Duncan's Flatmate

*Electron Version.

HALL OF FAME

USER GROUP MEETINGS

AUCKLAND: -meets 2nd Wednesday of the month, 7.30pm, Conference Rooms, Auckland Secondary Teachers College,60 Epsom Ave, Epsom. ph Dave 770-630 x 518(bus) or Kerry 695-355.
-North Shore: meets last Wednesday of the month, 7.30pm, Salvation Army Hall, Glenfield Road, Glenfield ph Chris 419-0543 (wk)

CHRISTCHURCH: -fortnightly workshops, Monday 6.30pm at Hagley High School. ph. Michael 582-267.

DUNEDIN: -first Thursday of the month. Phone Martin McDowall 42-031

HAWKES BAY: -Alternative months Hastings & Napier. Contact Kendall Napier 435-624, Bob Taradale 446-955, Mitch Hastings 778-235.

INVERCARGILL: -Phone Evan 330-444 Invercargill.

PALMERSTON NORTH:-Phone Tom Skinner 83-009(wk)

TAUPO: -meets every second Wednesday @ 7pm. Contact David 84-215 (h), 85-124 (wk).

TAURANGA: -every second Wednesday. ph. Chris & Jane 65-076.

TIMARU: -Phone Lloyd van der Krogt Timaru 61-412.

TOKOROA: -meets first Friday of each month, Tokoroa High School, 8pm. Phone Brian 67-625 Tokoroa.

WAIKATO: -first Monday of the month 7.30pm Waikato Tech Institute. Ph. Alison or John, Morrinsville 6695.

WELLINGTON: -meets last WEDNESDAY of the month, 7.30pm, Royal Society Lecture Theatre, Turnbull St, Thorndon.
-Workshops: second Thursday of month, Wellington College - ring Don 848-235 or Beryl 862-874 or Mike 785-437.

OTHER CENTRES: - let us know and we will publish details here.

THE NEWSLETTER:

CONTRIBUTIONS:-most welcome. Listings should be sent in on tape or disc. (Tape - 2 copies, one at 300 baud please). Please include written explanation of listing; text and graphics only material welcome too.

DEADLINES:- For a particular issue, the last day for material is the last day of the month prior to publication.

ADVERTISING:- Rates are $20 per half page (camera ready), deadline as above.

Copyright (c) BBC/ACORN COMPUTER USER GROUP NZ INC.

Published monthly except January and mailed to financial members.

MEMBERSHIP:

Membership of the users group is on payment of an annual subscription from April to March. For the 1985/6 year it is: $30.00, plus $5.00 for Auckland & Wellington members as a local branch levy, which entitles attendance at meetings as detailed above. Joining or renewing members get all the back issues of the current year - those joining from 1 October may have half the annual sub waived and only receive newsletters from 1 October.

BBC/ACORN COMPUTER USER GROUP NZ INC,
P.O.Box 9592, WELLINGTON.

Reproduced by BRYCE FRANCIS LTD, WELLINGTON.